# Qualitative Reasoning in a Traditionally Quantitative World: Modelling a Container Sytem

Steven van de Graaf, 2653345

*Abstract*—It has been orthodoxy to model dynamic systems and their behaviour strictly quantitatively. Since the 1980's, however, qualitative models of such dynamic systems have shown their worth in various contexts. In this report, a container system with a tap and a drain (which both can be opened and closed) is introduced. Firstly, the qualitative model of this system is presented, as well as the assumptions made concerning it. Next the implementation of this qualitative model is described, before finally,the state-graph it produced is presented and discussed.

## I. Introduction

Traditionally, dynamic systems and their behaviour have been modelled quantitatively, that is, using continuous quantities and mathematical formalisms to create a (parameterized) model of the system [2]. Quantitative models don't come without any drawbacks, however. For example, often-times, quantitative models are merely an idealization of the dynamic system they are supposed to represent. This might be due to a lack of understanding of the dynamic system itself, but often-times incomplete data plays a big part in this, too [4]. In such cases, for example, *soft* variables (e.g. "consumer satisfaction") may be introduced, which bring about a level of uncertainty. While independently these *soft* variables might not be problematic in terms of how representative the quantitative model is of the actual dynamic system it is supposed to represent, but when you have several of these *soft* variables, these uncertainties compound and they may indeed become problematic [2].

Since the early 1980's, however, qualitative approaches to the modelling of such dynamic systems have been put forward, which do not make use of continuous quantities, and allow for reasoning with little data. Qualitative reasoning about such systems is motivated by the observation that people can interact with and draw conclusions from the physical world without much (precise) data or complex mathematical formulas, but also by the observation that for the initial understanding of a problem, the formalization of a problem, and when interpreting the results of an experiment, scientists appear to already use qualitative reasoning [3]. Indeed, it is sometimes suggested that the existence of a qualitative model of a system is a prerequisite for a scientific, quantitative model, and not its result [1].

This report concerns itself with a container system (e.g. bathtubs, sinks, etc.), which can be filled with water (e.g. with a water tap), but also emptied (e.g. with a drain). A qualitative model of this system is first presented in Sec. II. Next, a discussion of the assumptions made regarding this system and the implications of these assumptions it put forward in Sec. III. This qualitative model (assumptions included) was implemented in code to allow for reasoning about the system and its behaviour, which is described in Sec. IV. Finally, the state-graph and trace that this implementation produces can be found in Sec. V.

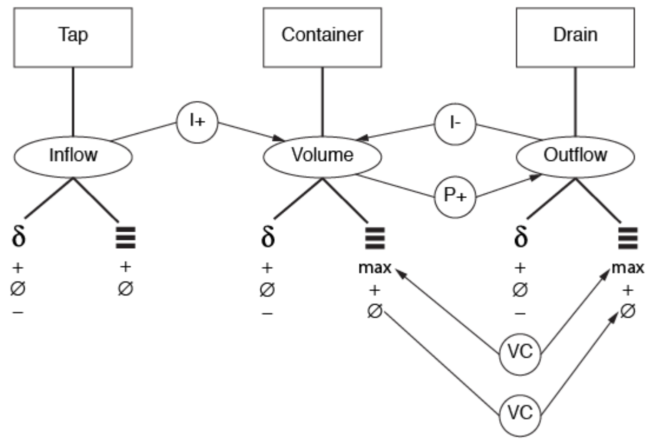


Fig. 1: Qualitative model of the container system

## II. Qualitative Model

As can be inferred from Fig. 1, our container system boasts 3 entities, *Tap*, *Container* and *Drain*, with 3 quantities, *Inflow*, *Volume* and *Outflow*, respectively. These 3 quantities all have the same derivatives (-, *0 and +)* and magnitudes (*0, + and max)*, with the exception of *Inflow*, which has no *max* value.

Moreover, there is a positive *Influence* relationship between *Inflow* and *Volume*, a negative *Influence* relationship between *Outflow* and *Volume*, and a positive *Proportional* relationship between *Volume* and *Outflow*.

Finally, there are two *Value Correspondence* relationships between *Volume* and *Outflow*, for magnitudes *max* and *0*, respectively.

## III. Assumptions

Both in the qualitative model, as its implementations, some assumptions about the container system were made, which are listed below:

1) The *0-Value Correspondence* between *Volume* and *Outflow* is unidirectional. This means that while there cannot be any outflow when there is no volume, there can be volume without any outflow (think of a drain stop).
2) When the *Inflow*, *Volume* and *Outflow* reach their maximum quantities, their derivatives cannot be positive. This means overflow of the container is not possible, for example.

3) When the *Inflow*, *Volume* and *Outflow* reach their minimum quantities, their derivatives cannot be negative. This means that there can't be sub-zero volume, for example.
4) The + magnitude values for the *Inflow*, *Volume* and *Outflow* are intervals, whereas the other values are point-values. This means that a state at an interval value does not have to change for a non-zero gradient, whereas for point values it does have to change.
5) Derivatives and magnitudes are continuous. This means that the *Volume*'s magnitude cannot go from *0* to *max* in one transition, without passing through *max* first, for example.
6) The *Tap* is extraneously controlled. This means that the tap can be opened (*0 to* + derivative change in *Inflow*) or closed (*0 to* - derivative change in *Inflow*) spontaneously at any time.

## IV. Implementation

The implementation of the qualitative model described in Sec. II, minding the assumptions listed in Sec. III, was done in the `Python 3.7`. Below a description of the data structures used to represent the states and state-transitions in-code is given, as well as an overview of the algorithm with which we created the state-graph presented in Sec. **??**.

### A. Data structures

*1) States:* In our implementation, states are represented as tuples, for example:

```
example_state = ('0','0','0','0','0','0'),
```

where the element at the first index is the magnitude of the *Inflow*, the element at the second index is the derivative of the *Inflow*, the element at the third index is the magnitude of the *Volume*, the element at the fourth index is the derivative of the *Volume*, the element at the fifth index is the magnitude of the *Outflow*, and finally, the element at the sixth index is the derivative of the *Outflow*.

Here, magnitude values can be either `'0'`, `'+'` or `'max'` and derivative values can be either `'-'`, `'0'` or `'+'`.

*2) State transitions:* Similarly to the states, as described previously, state transitions are also represented as tuples, for example:

```
example_state_transition = (
  ('0','0','0','0','0','0'),
  ('0','+','0','0','0','0')
),
```

where the element at the first index is the from-state tuple and the element at the second index is the to-state tuple of the state-transition.

### B. Algorithm

The algorithm we devised to create the state-graph presented in Sec. **??** consists of 5 steps, namely:

1) (Over)generation of states
2) Elimination of invalid states
3) (Over)generation of state-transitions
4) Elimination of invalid state-transitions
5) Creation of the state-graph

Below you'll find these steps explained in detail.

*1) (Over)generation of states:* In this step of the algorithm, given the quantities of the qualitative model and their magnitudes and derivatives, states are generated as all unique, possible combinations of these magnitudes and derivatives, not taking into account any of the dependencies or assumptions mentioned previously. As such, 486 states are generated in total for the qualitative model presented in Sec. II.

*2) Elimination of invalid states:* Given the various dependencies of our qualitative model – *I+(Inflow, Volume)*, *I-(Outflow, Volume)*, *P+(Volume, Outflow)*, *VC(Volume(max), Outflow(max))*, *VC(Volume(0), Outflow(0))* – as well as the assumptions listed in Sec. III, invalid states are eliminated.

This elimination was done by hard-coding all of the dependencies and assumptions as a comprehensive set of *if-else* statements. As such, after elimination of invalid states there were 20 valid states remaining.

*3) (Over)generation of state-transitions:* Given the remaining valid states, state-transitions are generated as unique, possible combinations of states. As such, 400 state-transitions are generated for our qualitative model.

*4) Elimination of invalid state-transitions:* Similarly to the elimination of invalid states (as presented in Sec. IV-B2), given the various dependencies of, as well as the assumptions made about the qualitative model, invalid state-transitions are eliminated by means of a comprehensive set of *if-else* statements. As such, after elimination of invalid state-transitions there were 75 valid state-transitions remaining.

*5) Creation of the state-graph:* Finally, given the valid states and state-transitions, the state-graph is created. This is done by adding a node to the graph for each valid state, as well as adding edges between these nodes for each valid state-transition.

## V. State-graph

Below in Fig. 2 the state-graph as produced by our algorithm (as presented in Sec. IV-B) for the qualitative model (as presented in Sec. II) is given.

In the state-graph, states are denoted by the magnitude and derivative values of their quantities. As such, for all three quantities you'll find their magnitude and derivative in square brackets (e.g. *[max, 0]*).

In the state-graph, you'll find that some states have loopy connections, meaning that they can transition from and to themselves. These loops are only possible when all derivatives are zero or when the quantities with a non-zero derivative are at an interval magnitude.

Also note that the assumption with respect to the erratic behaviour of the tap mentioned previously is clearly reflected in the state-graph. Take states 13 and 19, for example. State 19 boasts a steady inflow, with both an increasing volume and outflow. With state 13, the tap has been closed and the inflow is now decreasing.

Another thing to note is that some states (states 4, 8, 9, 12, 14 and 20, to be precise) only have outgoing edges and no incoming edges. This means that these states can never be reached from the starting state (see the `example_state` in Sec. IV-A1) where all quantities have 0 magnitude and 0 derivative. This is a direct result of our assumptions. Take state 8, for example. It cannot be reached from state 4, as it has non-zero derivatives for quantities at point-values. It can also not be reached from a state like `('0','0','+','+','+','+')` as it is invalid due to the dependencies of our qualitative model. As such, while these states are all valid according to our dependencies and assumptions, they don't really play a role in our state-graph. As such, Fig. 3 presents the state-graph with these inconsequential states removed, which now only contains 14 states and 54 state-transitions.

The last thing to note for both of these state-graphs is that the edges are labeled, which serve as an inter-state trace. The following labels have the following meanings:

- **E**rratic: When the derivative of a magnitude changes, without the magnitude itself changing
- **I**nterval: When a state doesn't change, because all its derivatives are zero or its quantities with a non-zero derivative are at an interval magnitude.
- **D**erivative: When one of the magnitudes of a state changes as a result of a non-zero derivative.

## VI. Conclusion

This report introduced the dynamic system of a container with a tap and a drain, which both can be opened and closed. First, the qualitative model for this system was presented in Sec. II, before listing the various assumptions made with respect to this system in Sec. III. Next, the algorithm used to create the state-graph shown in the previous section (Sec. V) is described in Sec. IV.

## References


[1] Bert Bredeweg and Peter Struss. Current topics in qualitative reasoning. *AI Magazine*, 24(4):13–13, 2003.

[2] Geoff Coyle. Qualitative and quantitative modelling in system dynamics: some research questions. *System Dynamics Review: The Journal of the System Dynamics Society*, 16(3):225–244, 2000.

[3] Kenneth D Forbus. Qualitative reasoning., 1997.

[4] Eric F Wolstenholme. Qualitative vs quantitative modelling: the evolving balance. *Journal of the Operational Research Society*, 50(4):422–428, 1999.


## Appendix

The source code of the algorithm described in Sec. IV is available for review at: https://github.com/sgraaf/QualitativeReasoning.

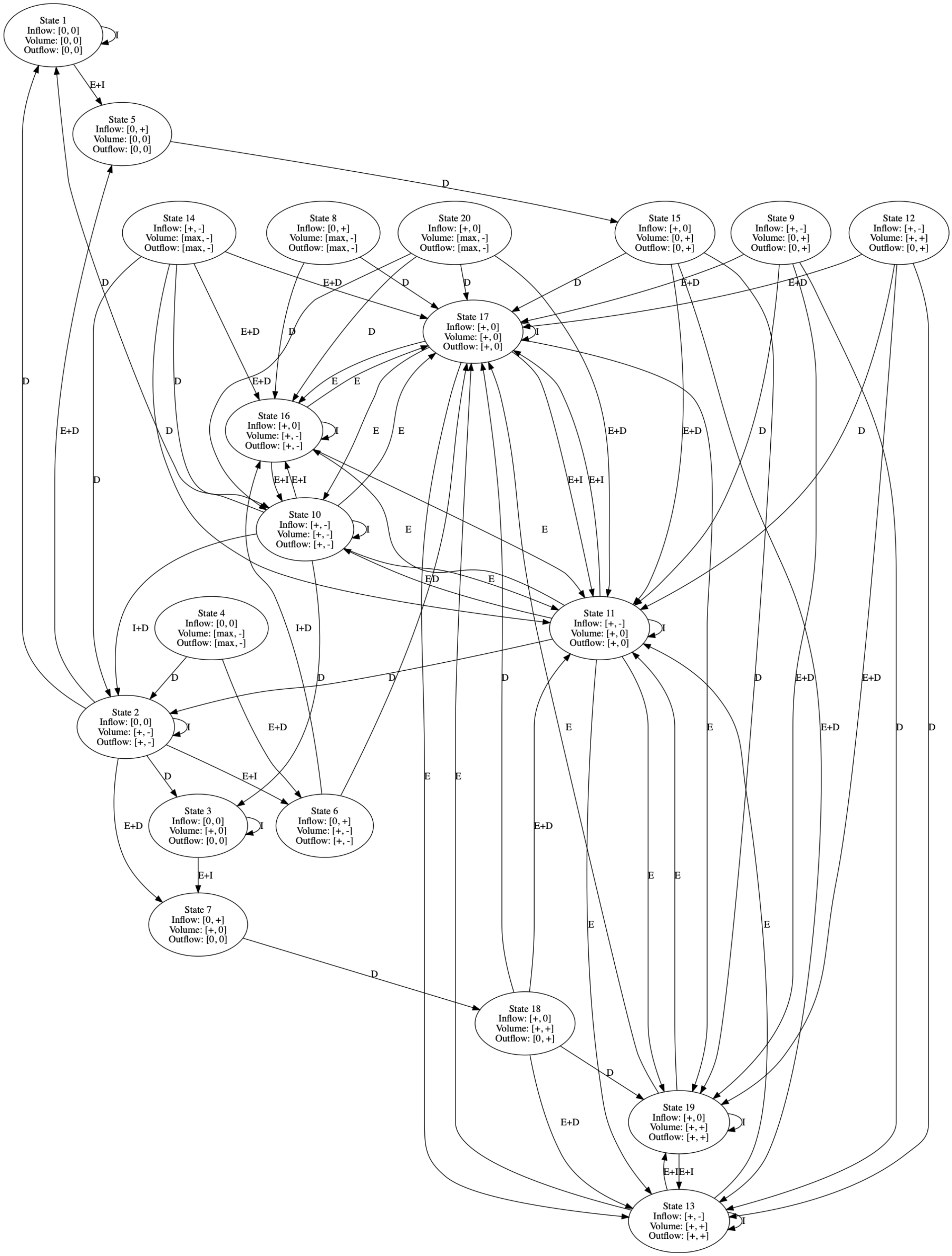


Fig. 2: The state-graph created by our algorithm for the qualitative model

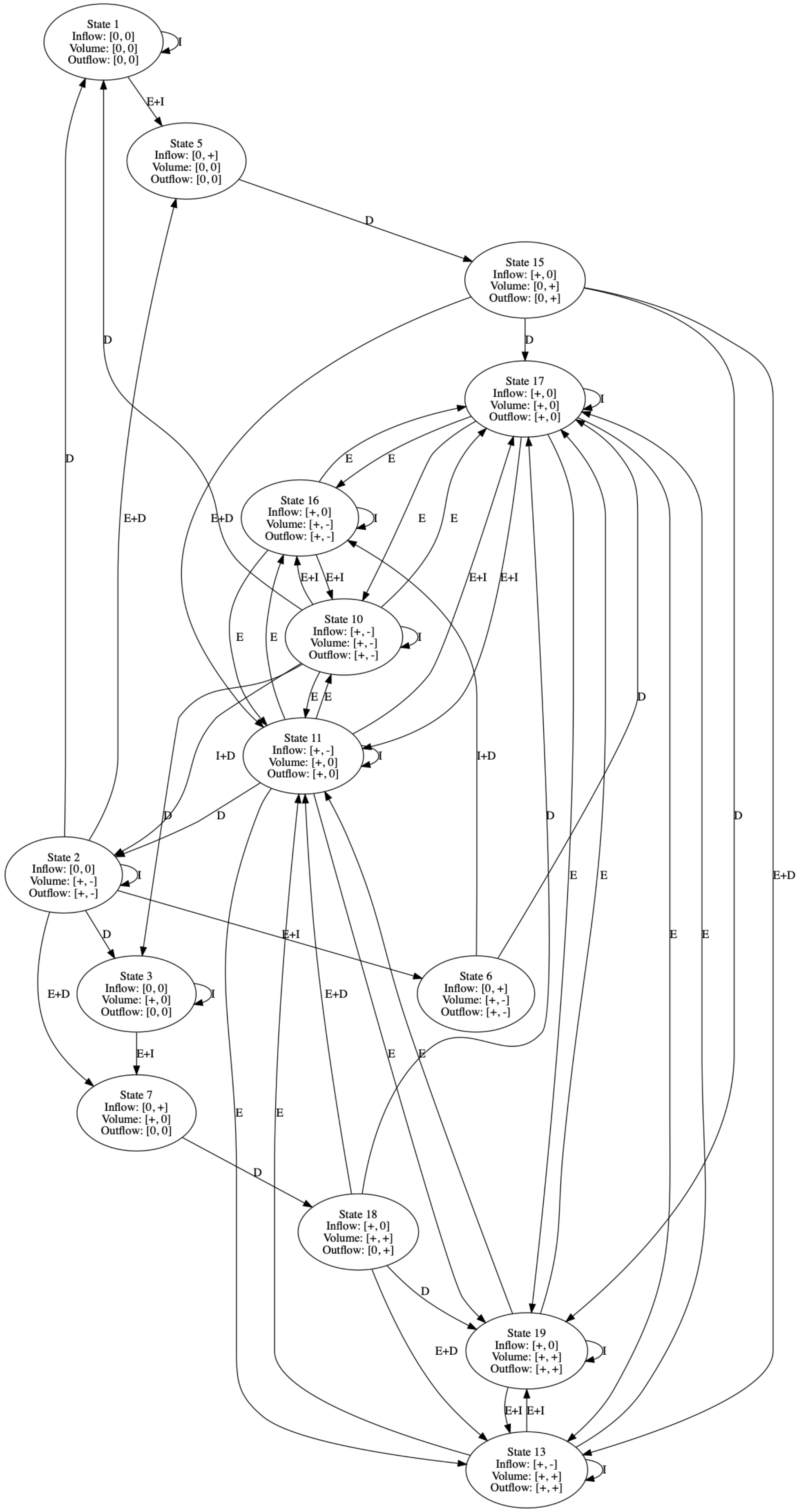


Fig. 3: The state-graph created by our algorithm for the qualitative model, with the inconsequential states removed